REMARKS

This amendment is filed in response to the non-final action mailed July 13, 2006. Applicant traverses the substantive grounds for rejection as discussed below. Reconsideration of the pending claims is requested.

In this response, all of the pending claims 1-31 are amended. No claims are canceled and no new claims are added.

The Abstract is replaced as requested by the Examiner.

Claims 1-31 were objected to for various informalities. The Applicant appreciates the Examiner's vigilant attention to the claim language. All of the claims have been carefully reviewed and amended as appropriate, and are now believed to obviate these objections. While the specific issues pointed out by the Examiner have been addressed, additional voluntary amendments also have been made for consistency, improved antecedent basis and the like. The changes all go to form or clarification rather than substance, except in the case of claim 17, as explained below.

Section 112, first paragraph

The phrase "implying receipt ..." is stricken from the claims, and thus the rejection of claims 1-16 and 31 based on enablement are obviated. In the claims, for example claim 1, this phrase was not intended to recite an additional method step, but rather to amplify on the "monitoring" step. To illustrate, the original language was, "monitoring each of the two-port memories to detect a non-empty condition, implying receipt of transferred data in the memory from the corresponding disk drive." The use of a comma rather than a semicolon was intended to indicate that the phrase was part of the monitoring step. As it was not essential to the claim, the language is deleted.

That said, Applicant would point out that the "implying receipt" phrase was included, and it is only mentioned here, because "implying receipt of the transferred data" indicated that the data indeed had been stored in the two-port memory to which it had been was "transferred". In this regard, the Examiner pointed out the absence of a distinct "storing step" (into the two-port memory) in a subsequent Section 112, second paragraph, rejection of claims 1 and 10. ("Data being *transmitted* to the memories is

recited, however the actual storage of said transmitted data is not previously set forth." Office action, page 5.[1]) A separate *storing* step intentionally was not recited here because it occurs by virtue of the transferring step. "In each case, the UDMA interface accepts data from the drive and pushes it into the FIFO on the drive's read strobe." Specification at [0020]. Not to belabor the point, but this feature is important to receiving data *asynchronously* from a bunch of independent drives.

Section 112, second paragraph

Claims 1-31 were rejected as being indefinite for various reasons. Again, all of the claims have been carefully reviewed and amended as appropriate, and are now believed to obviate these objections. One of them bears mention; "on-the-fly" is not believed to be indefinite. The phrase "on-the-fly" is explained in the specification (for example, see FIG. 3 "On-the-fly XOR -- Write Direction" and FIG. 4 "On-the-fly XOR -- Read Direction" and the corresponding portions of the specification. In addition, the specification includes the following passage:

"[0007] What is needed is a method and apparatus to effect synchronous data transfers, for example to and from a buffer, when the data transfers to and from the disk drives are actually asynchronous. The availability of synchronous data transfers would enable "on the fly" generation of redundancy information (in the disk write direction) and "on the fly" regeneration of missing data in the read direction (in the event of a disk failure) using the method as described in 6,237,052. US Pat. No. 6,237,052 is hereby incorporated herein" (emphasis added).

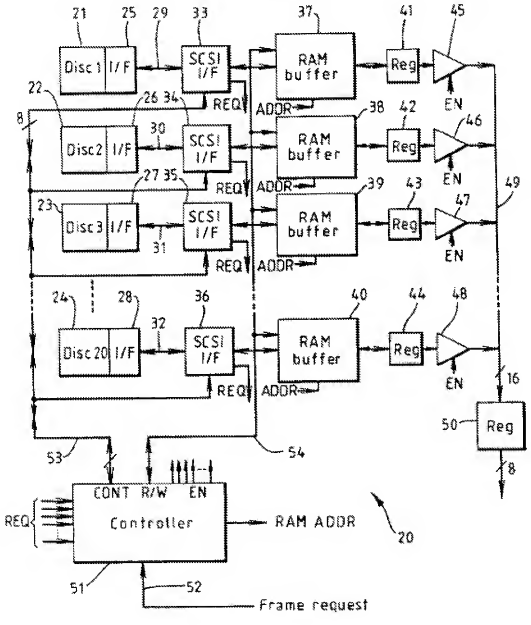
The phrase "on-the-fly" is discussed in detail in the Stolowitz patents of record; incorporation by reference was used here to reduce duplication of prior art. Roughly, the term means "without requiring a separate operation that would impose delay." The Examiner suggested the term "dynamic" (Office action, page 6). Applicant accepted that suggestion in the present amendments, recognizing that "on-the-fly" may perhaps appear idiomatic to some readers.

[1] The actual claim language is "transferred" rather than "transmitted". Applicant would agree with the Examiner that mere transmission does not always include receipt, but "transfer" is used in this case to imply successful receipt. Searby uses "transfer" in that sense.

Section 102(e) Anticipation

A. Claims 1, 3-5, 8, 10-12, 14, 16-17, 19, 21, 26-28, and 31 were all rejected as allegedly anticipated by Searby (US 5,765,186). Searby FIG. 2 is shown below.

FIG. 2



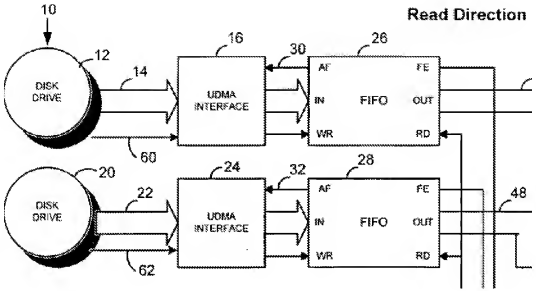
1. Searby synchronizes data transfer from the drives to the RAM buffer; Applicant's method of claim 1 leaves those transfers *asynchronous.*

Searby (FIG. 2) waits until all of the disc interfaces (SCSI 33-36) have data ready, as indicated by the REQ commands:

> "It will be appreciated by those possessed of the appropriate skills that because data transfer between the disc stores 21 to 24 and their respective second SCSI interfaces 33 to 36 is uneven and unsynchronised, the second SCSI interfaces 33 to 36 will receive their two bytes of data at different times. The controller 51 is arranged to utilise the buffering provided by the SCSI interface and to wait until all twenty of the second interfaces have output a request signal before outputting a write strobe signal over line 54 to the RAM buffers 37 to 40. The write strobe signal causes the RAM buffers 37 to 40 to receive a word of data from their respective second SCSI interfaces 33 to 36."

(Searby, Column 6, line 59 to column 7, line 3.) Similarly, the Abstract says, in pertinent part, "When data is ready for transfer a signal is sent from the interfaces 33 to 36 and once the controller has received signals from each of the interfaces it enables data to be transferred between the interfaces and respective RAM buffers."

Applicant's method of claim 1 – does NOT wait for each drive interface to signal "data ready." To the contrary, the data is transferred, per Applicant's claim 1, "*from the drive* into its corresponding two-port memory *using the timing signals provided by the respective drive*." Note the use of the singular form; the timing signals from each drive control its (and only its) corresponding two-port memory write. For example, in the illustrative embodiment of FIG. 1 (shown in part below), note the WR (write) and AF (almost full) signals between UDMA interface 16 and FIFO 26 are distinct from the WR and AF signals between interface 24 and its corresponding FIFO 28. Searby teaches synchronizing read data transfes from all the disc SCSI interfaces to a memory buffer; that is not what is described in the cited language of Applicant's claim 1.



For at least the foregoing reasons, claims 1 and 10 are not anticipated.

2. The claim step of "monitoring each of the two-port memories to detect a non-empty condition" is not taught in Searby. Searby actually "monitors" the RAM buffer (likely counts write commands or checks current RAM addresses) in order to transfer data to the highway 49 upon completion of a frame of video data.

The Examiner contends that Searby must be monitoring the RAM buffers of his system to "detect a non-empty condition," per claim 1, but Appplicant respectfully disagrees with this characterization of Searby. Specifically, the Examiner said that Searby "transfers the data to the data highway based on the determination that data is stored in the RAM buffers." Office action at page 8. The Examiner continued:

"Since all the buffers receive data substantially concurrently (i.e., all disk stores send data to the RAM buffers at the same time, which causes either all the buffers to contain data, or no buffers to contain data), each RAM is monitored to determine that data is either present (i.e., non-empty condition), or not present. Data present in the RAM buffers is an implied acknowledgment that data have been received from the disk stores. Note the system cannot transfer data to the highway until data is present in the RAM buffers; therefore the memories are monitored for this determination to occur)..." *Id.*

Searby explains that the data is written into all the RAM buffers ("substantially concurrently") by a common write strobe signal 54 from the controller 51. See Column 6, quoted above. The controller 51 also provides RAM addressing, see FIG. 2. Because the controller itself effects the writes to the RAM buffers (R/W strobe 54), it "knows" when they have received data from the drives. Further, as all RAMS receive data concurrently, monitoring any one of them would suffice to determine its status. This process in Searby is all synchronous and for that reason the controller really has no need at all to "monitor each of the two-port memories to detect a non-empty condition." That need arises in Applicant's method of claim 1 precisely because the two-port memories are receiving data *asynchronously* (each memory using the timing signals provided by the respective drive, as discussed above). For these additional reasons, claims 1 and 10 are not anticipated.

In that same vein, the steps of "waiting until all of the two-port memories indicate such a non-empty condition" and "then synchronously reading the ~~stored~~ transferred data from all of the two-port memories, thereby forming synchronous read data, and writing the synchronous read data into the buffer" are not anticipated. The Examiner again focused on the transfer from Searby's RAM buffers to the data highway. Searby's transfer from the RAM buffers to the highway though, is not "thereby forming synchronous read data." Rather, the data in Searby was *already* synchronized as it was written into the RAM buffer as discussed earlier.

Searby's controller is just waiting to collect a full frame of video data. "Once all of the data for the requested frame has been transferred to the RAM buffers 37 to 40 it can be output therefrom to the highway 49. Data is output from the RAM buffers 37 to 40 by applying a read strobe signal to the line 54 while simultaneously addressing the same location in each of the stores." See Column 7, lines 32 *et seq.* For these additional reasons, claims 1 and 10 are not anticipated.

The foregoing arguments also apply to all claims that depend from claim 1 or claim 10.

Claim 17

One feature of the invention of claim 17 is to bring the advantages of "on-the-fly" redundancy operations, first disclosed in Applicant's prior patents, to newer disk drive interfaces. Those operations require synchronous data transfers. "The availability of synchronous data transfers enables "on the fly" generation of redundancy information (FIG. 3) (in the disk write direction) and "on the fly" regeneration of missing data in the read direction (FIG. 4)." ABSTRACT. The specification also notes, "U.S. Pat. No. 6,237,052 B1 teaches that redundant data computations may be performed "On-The-Fly" during a synchronous data transfer. The combination of the three concepts: Synchronous Data Transfers, "On-The-Fly" redundancy, and the UDMA adapter using a FIFO per drive provides a high performance redundant disk array data path using a minimum of hardware." [0030]. Claim 17 is directed to an improved RAID controller that realizes that combination.

Searby discloses redundancy operations, including XOR, to provide corrected data in the event of an error, as noted by the Examiner. Redundancy for error correction in disk drives admittedly is not new. Nonetheless, Searby does not disclose circuitry to provide error correction on-the-fly in his system. To the contrary, see Searby FIG. 3 and the text at column 9 describing the requirement for a *delay shift register* 57:

> When an error is detected by the discs' own error checking facilities during the reading out of data therefrom a signal is sent via the control bus 53 to the controller 51 indicating which disc, and hence which data, contains the error. ...
>
> The data thus output from the registers 41 to 44 is input via highway 49 to a data regenerator unit 56 and a delay shift register 57. The data regenerator 56 performs a bit-wise exclusive-or operation, similar to that performed by the parity generator 55, on the data that it receives from highway 49. ...
>
> The data words are output from the delay shift register 57 in order until the parity word is reached. The parity word is discarded and the missing data word regenerated by the data regenerator 56 is output in its correct place in the data sequence. It should be noted that the delay shift register 57 is included to delay the outputting of data from the system while the data regenerator 56 calculates the missing data. The delay shift register 57 introduces a delay into the highway 49 corresponding to the

transmission of nineteen words of data on the highway. When there are no detected errors data is simply output unaltered via the delay shift register 57 to the register 50.

Searby at column 9, lines 24-60 (emphasis added). In the present case, no delay is incurred in these operations. The specification explains one preferred embodiment as follows:

[0032] In the Disk Write direction, data words are read from the buffer 350. Segments of these data words, e.g. see data paths 342, 344, are written to each of the drives. At this point, a logical XOR operation can be performed between the corresponding bits of the segments "on the fly". XOR logic 360 is arranged to compute the boolean XOR of the-corresponding bits of each segment, producing a sequence of redundant segments that are stored preliminarily in a FIFO 370, before transfer via UDMA interface 380 to a redundant or parity drive 390. Thus the XOR data is stored synchronously with the data segments. In other words, "On-The-Fly" generation of a redundant data pattern "snoops" the disk write process without adding any delays to it.

[0033] Turning now to FIG. 4 in the drawing, a similar diagram illustrates data flow in the read direction. The array of drives 300, corresponding interfaces 320 and FIFO memories 340 are shown as before. In the Disk Read direction, the XOR is computed across the data segments read from each of the data drives and the redundant drive. Thus, the data segments are input via paths 392 to XOR logic 394 to produce XOR output at 396. If one of the data drives has failed (drive 322 in FIG. 4), the result of the XOR computation at 394 will be the original sequence of segments that were stored on the now failed drive 322. This sequence of segments is substituted for the now absent sequence from the failed drive and stored along with the other data in the buffer 350. This substitution can be effected by appropriate adjustments to the data path. This data reconstruction does not delay the data transfer to the buffer, as more fully explained in my previous patents.

Claim 17 is amended to clarify that it includes the following language:

"one of the disk drive interfaces is designated for connection to a redundant disk drive for storing redundant data so that the synchronous read data includes redundant data; and

the first redundant data circuitry for regenerating missing data is coupled to all of the two-port memories to receive the synchronous read data, including the redundant data, and further is arranged to compute a redundant data operation across the

synchronous read data so as to provide corrected data in the event of a failed drive without delaying output of the read data."

Claim 17 and the claims that depend from it are not anticipated by Searby.

B. Claim 15 was rejected under Section 103(a) as unpatentable over Searby as applied to claim 10. (Office action, page 16.) Applicant does not contend that claim 15 is independently patentable; it is patentable for the reasons explained above with regard to claim 10, as amended.

C. Claims 6-7, 13, and 29-30 were rejected under Section 103(a) as unpatentable over Searby as applied to claims 1, 10 and 26 and further in view of Anderson (US Publication 2003/0200478 A1). Applicant does not contend that these claims are independently patentable; they are believed patentable for the reasons explained above with regard to the base claims, as amended.

D. Claims 2, 9, 20, and 22-25 were rejected under Section 103(a) as unpatentable over Searby as applied to claims 1, 17 and 26 and further in view of Stolowitz (US Pat. No. 6,018,778). (Office Action page 18.) Applicant does not contend that these claims are independently patentable; they are believed patentable for the reasons explained above with regard to the base claims, as amended.

E. Office action page 19. Claim 18 was rejected under Section 103(a) as unpatentable over Searby as applied to claim 17 and further in view of **Yamamoto** (U.S. Pat. No. 5,801,859).

In view of the above amendments and arguments the claims should be allowed.

The Commissioner is hereby authorized to charge shortages or credit overpayments to Deposit Account No. 19-4455.

Respectfully submitted,

NC2006 Assignee FBO Creditors LLC

By /Micah D. Stolowitz/
Micah D. Stolowitz
Registration No. 32,758

STOEL RIVES LLP
900 SW Fifth Avenue, Suite 2600
Portland, Oregon 97204-1268
Telephone: (503) 224-3380
Facsimile: (503) 220-2480
Attorney Docket No.: 44541/6:2